ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

Отчёт по лабораторной работе № 7

«410. Split Array Largest Sum»

Выполнил работу

Максудов Кирилл

Академическая группа J3111

Принято

Практик Вершинин Владислав

Санкт-Петербург

2024

Введение

Цель работы: дан целочисленный массив nums и целое число k. Необходимо разбить nums на k непустые подмассивы так, чтобы наибольшая сумма любого подмассива была минимальной.

Задачи:

1. Написать алгоритм
2. Подсчитать память
3. Измерить время работы алгоритма
4. Объяснить зачем нужен жадник.
5. Теоретическая подготовка

Использовал векторы, лямбда функцию, int, string, алгоритм бинарного поиска и тд. б

1. Реализация

После попытки решить через сортировку массива и составления подмассивов, но понял, что нельзя менять порядок элементов => подход невозможный, решил сменить подход и пришел к выводу, что минимальная сумма всегда лежит между максимальным элементом и суммой всех элементов, тк, самый маленькой суммой может быть подмассив из 1 самого большого элемента. В полученном промежутке ищу нужный минимум бинарным поиском, тк. значения отсортированы, каждый раз проверяя можно ли разбить на нужное количество подмассивов с таким минимумом.

По ходу решения столкнулся с проблемой невозможности создать обычную функцию в теле другой функции и воспользовался лямбда функцией.

1. Экспериментальная частью

Подсчет по памяти(только для циклов и сложных структур):

Используется только исходный вектор std::vector<int> nums => без его учета сложность по памяти O(1)

Подсчёт асимптотики (только для циклов и сложных структур):

Используется бинарный поиск со сложностью O(log(sum-max)) где sum – сумма всех элементов в векторе, а max – максимальный элемент этого вектора

Функция splitting перебирает все элементы вектора 1 раз => сложность O(n) где n - длинна вектора

Цикл поиска максимума и суммы вектора тоже перебирает все элементы по 1 разу – O(n)

Общая сложность O(n \* log(sum-max))

Изображение №1 – График зависимости времени от размера вектора

Таблица №1 - Подсчёт сложности реализованного алгоритма

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Размер вектора | 100 | 500 | 1000 | 2500 |
| Расчетное время, с | 0.00001810 | 0.00009869 | 0,00020453 | 0,00053509 |
| Время, c | 0.00001810 | 0.00006610 | 0.00013010 | 0.00061620 |

1. Заключение

В ходе выполнения работы мною был реализован алгоритм поиска. В минимальной возможной суммы элементов подмассива при разбиении на n элементов. В этой задаче используется жадный алгоритм в функции splitting, его использование заключается в постоянном добавлении новых, подряд идущих элементов в подмассив до тех пор, пока мы не превысим лимит.

ПРИЛОЖЕНИЕ А

Листинг кода файла lab-7.cpp

Изображение выглядит как текст, снимок экрана, программное обеспечение

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, программное обеспечение, дисплей

Автоматически созданное описание

ПРИЛОЖЕНИЕ B

Результат на сайте leetcode.com

Изображение выглядит как текст, Мультимедийное программное обеспечение, программное обеспечение, Графическое программное обеспечение

Автоматически созданное описание